

CLAIMS:

What is claimed is:

1. A method for providing scalable device driver services within a control system of a network processor services architecture comprising the steps of:

in response to determining a desired network processor functionality, loading a plurality of functional components that each provide a respective one of a plurality of network processor services;

providing at least one utility interposed between said plurality of functional components and an operating system (OS) of said control system, that provide an OS independent communication interface for said plurality of functional components; and

in response to a receipt of a packet at said control system, handling said packet utilizing one of said plurality of individual software components by first routing said packet through said utility, wherein said packet is decoded into a common code understandable by said OS and said one of said plurality of individual software components.

2. The method of Claim 1, wherein said loading step includes the step of loading external application programming interfaces (APIs), low level APIs, and physical transport interfaces of a device driver.

1 3. The method of Claim 2, further comprising the step
2 of loading a customer definable service component within
3 said external (APIs) that includes a customer desired
4 network service, which is operable within said network
5 processor services architecture.

1 4. The method of Claim 1, wherein said providing step
2 includes the step of providing a bi-directional
3 connection between said utility and said operating
4 system, one or more network processors, and each of said
5 functional components.

1 5. The method of claim 1, wherein said providing step
2 further comprises the step of linking a system services
3 utility to said operating system, wherein, said system
4 services utility operates to allow each of said
5 individual software coded components to communicate with
6 said OS.

1 6. The method of Claim 1, wherein said providing step
2 further provides a translation utility, which translates
3 all incoming and outgoing service requests into a common
4 network processor language to permit seamless connection
5 and correspondence between said one or more network
6 processors, said operating system, and each of said
7 functional components to enable handling of network
8 packets.

1 7. A computer program product for providing scalable
2 device driver services within a control system of a
3 network processor services architecture comprising:
4 a computer readable medium; and
5 program instructions on said computer readable
6 medium for:

7 implementing a plurality of individual software
8 components that each provide a respective one of a
9 plurality of network processor services;

10 providing at least one utility interposed between
11 said plurality of individual software components and an
12 operating system (OS) of said control system, that
13 provide an OS independent communication interface for
14 said plurality of individual software components; and

15 in response to a receipt of a packet at said control
16 system, handling said packet utilizing one of said
17 plurality of individual software components, by first
18 routing said packet through said utility, wherein said
19 packet is translated into a code understandable by said
20 OS and said one of said plurality of individual software
21 components.

1 8. The computer program product of Claim 7, wherein
2 program instructions for said loading step includes
3 program instructions for loading external application
4 programming interfaces (APIs), low level APIs, and
5 physical transport interfaces of a device driver.

1 9. The computer program product of Claim 8, further
2 comprising program instructions for loading a customer
3 definable service component within said external APIs
4 that includes a customer desired network service, which

5 is operable within said network processor services
6 architecture.

1 10. The computer program product of Claim 7, wherein
2 said program instructions for said providing step
3 includes instructions for providing a bi-directional
4 connection between said utility and said operating
5 system, one or more network processors, and each of said
6 functional components.

1 11. The computer program product of claim 7, wherein
2 said program instructions for said providing step further
3 comprises instructions for linking a system services
4 utility to said operating system, wherein, said system
5 services utility operates to allow each of said
6 individual software coded components to communicate with
7 said OS.

1 12. The computer program product of Claim 7, wherein
2 said program instructions for said providing step further
3 includes program instructions for a translation utility,
4 which translates all incoming and outgoing service
5 requests into a common network processor language to
6 permit seamless connection and correspondence between
7 said one or more network processors, said operating
8 system, and each of said functional components to enable
9 handling of network packets.

1 13. A control system of a network processor services
2 architecture comprising:

3 a plurality of individually loadable functional
4 components within a device driver of said control system
5 that each represents a network processor service;

6 at least one utility for enabling each of said
7 functional components to communicate with an operating
8 system (OS) of said control system, wherein said utility
9 responsive to a receipt of a request by said OS to
10 perform a network processor function, translates said
11 request into a call of a particular one of said plurality
12 of functional components that administers said network
13 processor services of one or more network processors; and

14 processing hardware that executes said OS, said
15 utility and said functional components.

1 14. The system of Claim 13, wherein said plurality of
2 functional components includes external application
3 programming interfaces (APIs), low level APIs, and
4 physical transport interfaces of a device driver.

1 15. The system of Claim 14, further comprising a
2 customer definable service component within said external
3 (APIs) that includes a customer desired network service,
4 which is operable within said network processor services
5 architecture.

1 16. The system of Claim 13, wherein said utility
2 includes a system services utility coupled to said
3 operating system that operates to allow each of said
4 individual software coded components to communicate with
5 said OS.

1 17. The system of Claim 13, wherein said utility
2 includes a translation utility, which translates all
3 incoming and outgoing service requests into a common
4 network processor language to permit seamless connection
5 and correspondence between said one or more network
6 processors, said operating system, and each of said
7 functional components to enable handling of network
8 packets.

1 18. A network processing services architecture
2 comprising:
3 one or more network processors; and
4 a control system, coupled to said one or more network
5 processors, having a scalable device driver for
6 controlling interactions with said one or more network
7 processors, wherein said scalable device driver is
8 comprised of a plurality of individual functional
9 software components including a plurality of functional
10 elements for network processing and a conversion utility
11 to enable compatibility between said plurality of
12 functional elements and said one or more network
13 processors to effectively communicate and interact within
14 any processor type and OS to effectuate the transfer of
15 frames and control information between said control
16 system and said one or more network processors.

17 19. The network processing services architecture of
18 Claim 18, further comprising a switch fabric coupled to
19 said one or more network processors.

1 20. A device driver program for a distributed
2 processing system comprising:

3 a plurality of system services programs providing
4 high level bi-directional communication between a common
5 code program and each of a network transport manager, a
6 network resource manager and a control message formatter,
7 said network transport manager providing low level
8 bi-directional communication to a physical transport
9 processor, said physical transport processor providing
10 primitive bi-directional communication to one or more
11 network processors.